Formal Languages and Compilers – JFlex & Cup
9 September 2009

Using the JFLEX lexer generator and the CUP parser generator, realize a Java program that is able to recognize and execute a simplified version of the C language

Input Language

The input file starts with an header; the header comprises a list with an uneven number of words (at least 3). Each word is separated from the others by a "**;**" symbol. Each word begins with a set of lowercase letters (minimum 3 lowercase letters, maximum 6) or with an integer number in the range 213 – 3720; then in the word there is or an uneven number of "-" symbol or an even number of "?" symbol; the word can optionally be ended with a valid IP address.

The "%" symbol separates the header from the simplified C program.

Please notice that the simplified C source can also be formed by 0 instructions. The program starts with a section with varibles declaration (the only allowed types are int and float); variables can be initialized in the declaration (i.e. *float a=3.2 , x=3;*)

The section of variables declaration is followed by a set of C instructions; only 3 different instructions are allowed and can be in any order and number:

- if, it has the standard C syntax without boolean operators (*&& || !*); the only comparison operators allowed are: > < ==. It is not possible to have nested if statements and within the if blocks is not possible to define new variables. In order to evauate the if condition (and execute or not its action block inherited attributes MUST be used)
- Assignments and algebrical operations
- The print operation that receives as parameter a varible (i.e. *print(x); ..it prints the value of the variable x*)

It is additionally required to perform type-checking for the assignments, the algebric operations and the comparison operations within if statements (i.e. *Assignin a float value to an integer variable results in a type error, all float values MUST have a decimal part as in float b=2.0*). If there is such an error the program must print an error message ("TYPE ERROR") and terminate.

NB: In the parser code no global variables are allowed with the exception of one or two hash tables in order to manage variables types and values

Goal

The program must be capable to recognize such input language and execute its instructions.

Examples:

Example A

abc-; abcd---120.130.2.27; 213????;

%
int a=3.2, b=0, c;

Output Example A

TYPE ERROR!:

Example B

```
xyz-----; 3790---123.255.255.0; 300?????;
%
int a=2, b, c=8;
float b=6.3, e=8.9;
a=a+5;
if (a+2>6-c){
        d=-(d*2.0);
        print(d);
}
d=d*2;
print(d);
```

Output Example B

-12.6
TYPE ERROR!:

Example C

xyz-;xyz-;xyz-;

```
%
int a=6;
float b,c,d;

b=3.2*2.0;
if(b<5.0){
        print(b);
}else{
        a=0;
        print(a);
}
a=1+3*2;
print(a);
if(b<5){
        print(b);
}
```

Output Example C

0
7
TYPE ERROR!: